

# Chapter 28

## Data Mining Concepts



Addison-Wesley  
is an imprint of

PEARSON

# Definitions of Data Mining

- The discovery of new information in terms of patterns or rules from vast amounts of data.
- The process of finding interesting structure in data.
- The process of employing one or more computer learning techniques to automatically analyze and extract knowledge from data.

# Data Warehousing

- The data warehouse is a historical database designed for decision support.
- Data mining can be applied to the data in a warehouse to help with certain types of decisions.
- Proper construction of a data warehouse is fundamental to the successful use of data mining.

# Knowledge Discovery in Databases (KDD)

- Data mining is actually one step of a larger process known as **knowledge discovery in databases (KDD)**.
- The KDD process model comprises six phases
  - Data selection
  - Data cleansing
  - Enrichment
  - Data transformation or encoding
  - Data mining
  - Reporting and displaying discovered knowledge

# Goals of Data Mining and Knowledge Discovery (PICO)

## ■ **Prediction:**

- Determine how certain attributes will behave in the future.

## ■ **Identification:**

- Identify the existence of an item, event, or activity.

## ■ **Classification:**

- Partition data into classes or categories.

## ■ **Optimization:**

- Optimize the use of limited resources.

# Types of Discovered Knowledge

- Association Rules
- Classification Hierarchies
- Sequential Patterns
- Patterns Within Time Series
- Clustering

# Association Rules

- Association rules are frequently used to generate rules from **market-basket data**.
  - A market basket corresponds to the sets of items a consumer purchases during one visit to a supermarket.
- The set of items purchased by customers is known as an **itemset**.
- An **association rule** is of the form  $X \Rightarrow Y$ , where  $X = \{x_1, x_2, \dots, x_n\}$ , and  $Y = \{y_1, y_2, \dots, y_n\}$  are sets of items, with  $x_i$  and  $y_j$  being distinct items for all  $i$  and all  $j$ .
  - For an association rule to be of interest, it must satisfy a minimum support and confidence.

# Association Rules

## Confidence and Support

### ■ Support:

- The minimum percentage of instances in the database that contain all items listed in a given association rule.
- Support is the percentage of transactions that contain all of the items in the itemset, LHS U RHS.

### ■ Confidence:

- Given a rule of the form  $A \Rightarrow B$ , rule confidence is the conditional probability that B is true when A is known to be true.
- Confidence can be computed as
  - $\text{support}(\text{LHS} \cup \text{RHS}) / \text{support}(\text{LHS})$



# Generating Association Rules

- The general algorithm for generating association rules is a two-step process.
  - Generate all itemsets that have a support exceeding the given threshold. Itemsets with this property are called **large** or **frequent itemsets**.
  - Generate rules for each itemset as follows:
    - For itemset  $X$  and  $Y$  a subset of  $X$ , let  $Z = X - Y$ ;
    - If  $\text{support}(X)/\text{Support}(Z) > \text{minimum confidence}$ , the rule  $Z \Rightarrow Y$  is a valid rule.

# Reducing Association Rule Complexity

- Two properties are used to reduce the search space for association rule generation.
  - **Downward Closure**
    - A subset of a large itemset must also be large
  - **Anti-monotonicity**
    - A superset of a small itemset is also small. This implies that the itemset does not have sufficient support to be considered for rule generation.

# Generating Association Rules: The Apriori Algorithm

- The **Apriori algorithm** was the first algorithm used to generate association rules.
  - The Apriori algorithm uses the general algorithm for creating association rules together with downward closure and anti-monotonicity.

# Generating Association Rules: The Sampling Algorithm

- The **sampling algorithm** selects samples from the database of transactions that individually fit into memory. Frequent itemsets are then formed for each sample.
  - If the frequent itemsets form a superset of the frequent itemsets for the entire database, then the real frequent itemsets can be obtained by scanning the remainder of the database.
  - In some rare cases, a second scan of the database is required to find all frequent itemsets.

# Generating Association Rules: Frequent-Pattern Tree Algorithm

- The **Frequent-Pattern Tree** Algorithm reduces the total number of candidate itemsets by producing a compressed version of the database in terms of an FP-tree.
- The FP-tree stores relevant information and allows for the efficient discovery of frequent itemsets.
- The algorithm consists of two steps:
  - Step 1 builds the FP-tree.
  - Step 2 uses the tree to find frequent itemsets.

# Step 1: Building the FP-Tree

- First, frequent 1-itemsets along with the count of transactions containing each item are computed.
- The 1-itemsets are sorted in non-increasing order.
- The root of the FP-tree is created with a “null” label.
- For each transaction T in the database, place the frequent 1-itemsets in T in sorted order. Designate T as consisting of a head and the remaining items, the tail.
- Insert itemset information recursively into the FP-tree as follows:
  - if the current node, N, of the FP-tree has a child with an item name = head, increment the count associated with N by 1 else create a new node, N, with a count of 1, link N to its parent and link N with the item header table.
  - if tail is nonempty, repeat the above step using only the tail, i.e., the old head is removed and the new head is the first item from the tail and the remaining items become the new tail.

# Step 2: The FP-growth Algorithm For Finding Frequent Itemsets

Input: Fp-tree and minimum support, mins

Output: frequent patterns (itemsets)

procedure FP-growth (tree, alpha);

Begin

if tree contains a single path P then

for each combination, beta of the nodes in the path

generate pattern (beta U alpha)

with support = minimum support of nodes in beta

else

for each item, i, in the header of the tree do

begin

generate pattern beta = (i U alpha) with support = i.support;

construct beta's conditional pattern base;

construct beta's conditional FP-tree, beta\_tree;

if beta\_tree is not empty then

FP-growth(beta\_tree, beta);

end;

End;

# Generating Association Rules: The Partition Algorithm

- Divide the database into non-overlapping subsets.
- Treat each subset as a separate database where each subset fits entirely into main memory.
- Apply the Apriori algorithm to each partition.
- Take the union of all frequent itemsets from each partition.
- These itemsets form the global candidate frequent itemsets for the entire database.
- Verify the global set of itemsets by having their actual support measured for the entire database.



# Complications seen with Association Rules

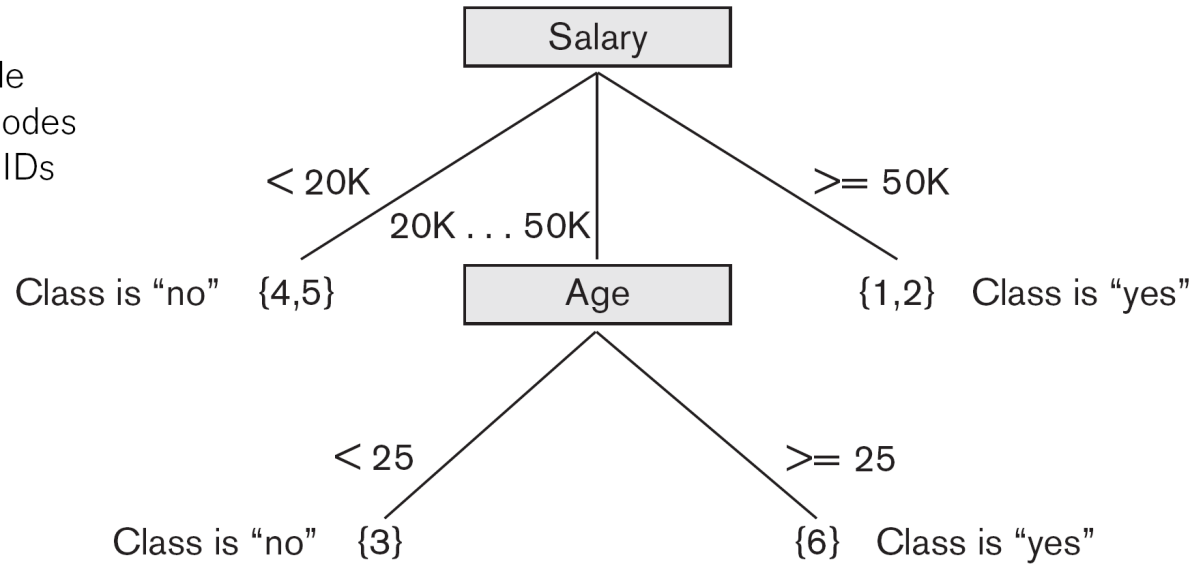
- The cardinality of itemsets in most situations is extremely large.
- Association rule mining is more difficult when transactions show variability in factors such as geographic location and seasons.
- Item classifications exist along multiple dimensions.
- Data quality is variable; data may be missing, erroneous, conflicting, as well as redundant.

# Classification

- **Classification** is the process of learning a model that is able to describe different classes of data.
- Learning is **supervised** as the classes to be learned are predetermined.
- Learning is accomplished by using a training set of pre-classified data.
- The model produced is usually in the form of a decision tree or a set of rules.

**Figure 28.7**

Decision tree based on sample training data where the leaf nodes are represented by a set of RIDs of the partitioned records.



# An Example Rule

- Here is one of the rules extracted from the decision tree of Figure 28.7.

IF 50K > salary >= 20K

AND age >=25

THEN class is “yes”

# Clustering

- Unsupervised learning or clustering builds models from data without predefined classes.
- The goal is to place records into groups where the records in a group are highly similar to each other and dissimilar to records in other groups.
- The **k-Means** algorithm is a simple yet effective clustering technique.

# Additional Data Mining Methods

- **Sequential pattern analysis**
- **Time Series Analysis**
- **Regression**
- **Neural Networks**
- **Genetic Algorithms**

# Sequential Pattern Analysis

- Transactions ordered by time of purchase form a sequence of **itemsets**.
- The problem is to find all **subsequences** from a given set of sequences that have a minimum support.
- The sequence  $S_1, S_2, S_3, ..$  is a predictor of the fact that a customer purchasing itemset  $S_1$  is likely to buy  $S_2$  , and then  $S_3$ , and so on.

# Time Series Analysis

- **Time series** are sequences of events. For example, the closing price of a stock is an event that occurs each day of the week.
- Time series analysis can be used to identify the price trends of a stock or mutual fund.
- Time series analysis is an extended functionality of **temporal** data management.



# Regression Analysis

- A **regression equation** estimates a **dependent** variable using a set of **independent** variables and a set of constants.
- The independent variables as well as the dependent variable are numeric.
- A regression equation can be written in the form  $Y=f(x_1, x_2, \dots, x_n)$  where  $Y$  is the dependent variable.
- If  $f$  is linear in the domain variables  $x_i$ , the equation is call a **linear regression equation**.

# Neural Networks

- A **neural network** is a set of interconnected nodes designed to imitate the functioning of the brain.
- **Node connections** have **weights** which are modified during the learning process.
- Neural networks can be used for supervised learning and unsupervised clustering.
- The output of a neural network is **quantitative** and not easily understood.

# Genetic Learning

- **Genetic learning** is based on the theory of evolution.
- An initial population of several candidate solutions is provided to the learning model.
- A fitness function defines which solutions survive from one generation to the next.
- **Crossover, mutation and selection** are used to create new population elements.

# Data Mining Applications

- **Marketing**

- Marketing strategies and consumer behavior

- **Finance**

- Fraud detection, creditworthiness and investment analysis

- **Manufacturing**

- Resource optimization

- **Health**

- Image analysis, side effects of drug, and treatment effectiveness

# Recap

- Data Mining
- Data Warehousing
- Knowledge Discovery in Databases (KDD)
- Goals of Data Mining and Knowledge Discovery
- Association Rules
- Additional Data Mining Algorithms
  - Sequential pattern analysis
  - Time Series Analysis
  - Regression
  - Neural Networks
  - Genetic Algorithms